Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Boundedness for Unions of Conjunctive Regular Path Queries over Simple Regular Expressions

Diego Figueira[1], **S. Krishna**[2], **Om Swostik Mishra**[2], **Anantha Padmanabha**[3]
[1]Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Talence, France
[2]Indian Institute of Technology Bombay, Mumbai, India
[3]Indian Institute of Technology Madras, Chennai, India
diego.figueira@cnrs.fr, {krishnas, 21b090022}@iitb.ac.in, ananthap@cse.iitm.ac.in

Abstract

The problem of checking whether a recursive query can be rewritten as query without recursion is a fundamental reasoning task, known as the boundedness problem. Here we study the boundedness problem for Unions of Conjunctive Regular Path Queries (UCRPQs), a navigational query language extensively used in ontology and graph database querying.

The boundedness problem for UCRPQs is ExpSpace-complete. Here we focus our analysis on UCRPQs using simple regular expressions, which are of high practical relevance and enjoy a lower reasoning complexity.

We show that the complexity for the boundedness problem for this UCRPQs fragment is Π_2^p-complete, and that an equivalent bounded query can be produced in polynomial time whenever possible.

When the query turns out to be unbounded, we also study the task of finding an equivalent maximally bounded query, which we show to be feasible in Π_2^p. As a side result of independent interest stemming from our developments, we study a notion of succinct finite automata and prove that its membership problem is in NP.

☞ This pdf contains internal links: clicking on a notion leads to its *definition*.[1]

1 Introduction

Regular Path Queries (RPQ) and its extension under conjunctions, known as Conjunctive RPQ (CRPQ) are a well-known generalization of conjunctive queries with a mild form of recursion, which are extensively used for querying knowledge bases and graph-structured datasets. In particular, CRPQs are part of SPARQL, which is the W3C standard for querying RDF data, including well known knowledge bases such as DBpedia and Wikidata. In particular, RPQs are extensively used according to recent studies (Bonifati, Martens, and Timm 2019; Malyshev et al. 2018). More generally, CRPQs are basic building blocks for querying graph databases (Angles et al. 2017; Baeza 2013).

As knowledge bases become larger, reasoning about queries (e.g. for optimization) becomes increasingly important. In this vein, many static analysis aspects of CRPQ have been studied, starting with the seminal papers on containment of CRPQs (Florescu, Levy, and Suciu 1998; Calvanese et al. 2000) (*i.e.*, whether the results of a query q_1 always contain those of q_2), which spawned many subsequent works. In particular, for queries with recursion such as CRPQ, a basic reasoning task is whether the recursion can be bounded. In other words, whether a given query can be equivalently written as a finite union of conjunctive queries (UCQ), known as the BOUNDEDNESS problem. UCQs form the core of most systems for data management and ontological query answering, and, in addition, are the focus of advanced optimization techniques. The BOUNDEDNESS problem has garnered attention, in particular for Datalog programs. For ontology-mediated query answering (OMQA), the problem is known as the "FO-rewritability" or "UCQ-rewritability" problem, which typically takes, as ontology-mediated query, a conjunctive query and some TBox formulated in some description logic (see, *e.g.*, (Bienvenu et al. 2016)). In this sense, the current work can be seen as a preliminary study for investigating FO-rewritability of ontology-mediated CRPQ queries.

Boundedness of CRPQs has been shown to be decidable, ExpSpace-complete (Barceló, Figueira, and Romero 2019), that is, as hard (or easy) as the containment problem for CRPQs (Calvanese et al. 2000). This holds also for the extension with union (UCRPQ) and two-way navigation (UC2RPQ). Further, whenever a query is bounded the equivalent UCQ may be of triply exponential size, and hence not directly amenable to an optimization procedure.

However, the lower bounds for boundedness —and generally for most static analysis problems on CRPQ— use rather complex regular expressions that are hardly used in practice. This has raised the question of the status of these problems when restricted to simpler basic regular expressions.

In fact, recent studies reveal that most CRPQ use extremely simple regular expressions of the form a^* or $a_1 + \cdots + a_n$. Studies on query-logs (Bonifati, Martens, and Timm 2019; Malyshev et al. 2018) show that these expressions cover more than 98% of all RPQ queries made on Wikidata and more than 46% of the queries made in DBpedia —see (Figueira et al. 2020a, Table 1) for more details.

A line of research was then established into understanding the difficulty of treating CRPQ over such simple regular expressions. It has been shown that over simpler expressions the situation improves drastically for the containment problem (Figueira et al. 2020a, Table 2) and the "semantic tree-

[1]https://ctan.org/pkg/knowledge

width problem", that is, whether the query is equivalent to a query of low tree-width (Figueira and Morvan 2023, §6).

Contributions We study the boundedness problem for UCRPQs restricted to a class of simple regular expressions. Such regular expressions can be either of the form w^*, where w is a word, or any regular expression which does not use Kleene star. We even allow for "succinct exponentiation" of the form w^n and $w^{\leq n}$ where n is encoded in binary.

For such queries we show that the boundedness problem is Π_2^p-complete. Both the upper and lower-bound proofs are non-trivial, but our main technical contribution is the upper bound (Section 5). Whenever the UCRPQ is bounded, it can be written as a UCRPQ of polynomial size which does not contain expressions with Kleene star.

As a necessary ingredient for the Π_2^p upper bound proof, in Section 4 we introduce a notion of "succinct automata" and use its membership checking problem, which we prove to be in NP, to solve the boundedness problem. Succinct automata are classic finite automata extended with transitions of the form $p \xrightarrow{w^n} p'$ to indicate a path of transitions from p to p' reading the word w^n, where n is represented in binary. These automata could be of independent interest when looking for succinct models of computation.

We also consider the related problem of bounding the query "by letters", *i.e.* whether the query is equivalent to one not making any recursion on a given subset of letters (Section 7). In this setting, the boundedness problem corresponds to bounding *all* letters occurring in the query. We show that there is a notion of "maximally bounded" query, *i.e.*, a query that is bounded in a maximum number of letters which is also computable within the Π_2^p bound.

Finally, in Section 6 we prove that the Π_2^p lower bound holds even for CRPQs with very simple regular expressions of the form a^* or a.

Organization Section 2 describes the preliminaries needed for the paper. Then we formally state our main results in Section 3. We dedicate Section 4 to the discussed side result on "succinct automata" and Section 5 to the main upper bound result. Section 6 elaborates on the lower bound. Section 7 delves into the problem of bounding a query "by letter". We conclude with Section 8.

2 Preliminaries

Let $\mathbb{A}$ be a finite alphabet. A *graph database* over $\mathbb{A}$ is a finite edge-labelled directed graph $G = (V, E)$ over $\mathbb{A}$, where V is a finite set of vertices and $E \subseteq V \times \mathbb{A} \times V$ is the set of labelled edges. We write $u \xrightarrow{a} v$ to denote an edge $(u, a, v) \in E$. We write $\mathit{Im}(f)$ to denote the image of any function f.

We denote $\mathbb{A}^*$ as the set of all finite words over $\mathbb{A}$ and ε to denote the empty word. A *path* from u to v in a graph database $G = (V, E)$ over alphabet $\mathbb{A}$ is a (possibly empty) sequence of edges $\sigma = v_0 \xrightarrow{a_1} v_1, v_1 \xrightarrow{a_2} v_2, \ldots, v_{k-1} \xrightarrow{a_k} v_k$ where each $v_i \xrightarrow{a_{i+1}} v_{i+1} \in E$, $v_0 = u, v_k = v$. The *label* of the path σ from u to v is the word $a_1 a_2 \ldots a_k$ of edge labels seen along the path. When $k = 0$, the label of the path is ε; this is called the "empty path", and there is always an empty path from u to u, for every vertex u.

Unless otherwise stated, we assume that regular languages $L \subseteq \mathbb{A}^*$ are encoded via regular expressions. A *regular path query* (*RPQ*) over $\mathbb{A}$ is a regular language L given as a regular expression. The semantics of L on a graph database $G = (V, E)$ over $\mathbb{A}$, written $L(G)$, is the set of pairs $(u, v) \in V \times V$ such that there is a directed path from u to v in G whose label belongs to L.

A *Conjunctive RPQ* (*CRPQ*) over $\mathbb{A}$ is an expression $q := \exists \bar{x}\big((y_1 \xrightarrow{L_1} z_1) \wedge (y_2 \xrightarrow{L_2} z_2) \wedge \ldots \wedge (y_m \xrightarrow{L_m} z_m)\big)$ where each L_i is an RPQ over $\mathbb{A}$. We call each $y_i \xrightarrow{L_i} z_i$ an *atom*. Further, $\bar{x}$ is a tuple of variables contained in $\{y_1, z_1, \ldots, y_m, z_m\}$ and the variables of q not contained in $\bar{x}$ are the *free variables* of q. A query with no free variables is called *Boolean*.

To simplify the definitions and technical developments we shall henceforth assume that all queries we deal with are Boolean. However, all our results and bounds hold for non-Boolean queries as well (modulo some slightly more cluttered definitions and proofs).

In the context of graph databases a *conjunctive query* (*CQ*) over $\mathbb{A}$ can be defined as a CRPQ over $\mathbb{A}$ of the form $\exists \bar{z} \bigwedge_{1 \leq i \leq m} x_i \xrightarrow{L_i} y_i$ where each L_i is a regular expression of the form a for some letter $a \in \mathbb{A}$, which denotes the singleton set $\{a\}$.

Given Boolean CQs q, q', a *homomorphism* from q to q' is a mapping $h : \mathit{vars}(q) \to \mathit{vars}(q')$ such that for every atom $x \xrightarrow{a} y$ of q, we have that $h(x) \xrightarrow{a} h(y)$ is an atom of q'. We write $q \xrightarrow{\mathit{hom}} q'$ when such homomorphism exists and $h : q \xrightarrow{\mathit{hom}} q'$ to denote that h is one such homomorphism. A *homomorphism* $q \xrightarrow{\mathit{hom}} G$ on a graph database $G = (V, E)$ is defined analogously, as a mapping $h : \mathit{vars}(q) \to V$ such that for every atom $x \xrightarrow{a} y$ of q, we have $h(x) \xrightarrow{a} h(y) \in E$.

Expansions Any UCRPQ can be equivalently seen as an infinite union of CQs. We define formally the shape of such CQs, which we call expansions.

CRPQs with *equality atoms* are queries of the form $q(\bar{x}) = \delta \wedge I$, where δ is a CRPQ (without equality atoms) and I is a conjunction of equality atoms of the form $x = y$. We define the binary relation $=_q$ over $\mathit{vars}(q)$ to be the reflexive-symmetric-transitive closure of the binary relation $\{(x, y) \mid x = y \text{ is an equality atom in } q\}$. In other words, we have $x =_q y$ if the equality $x = y$ is forced by the equality atoms of q. Note that every CRPQ with equality atoms $q(\bar{x}) = \delta \wedge I$ is equivalent to a CRPQ without equality atoms $q^{\approx}$, which is obtained from q by collapsing each equivalence class of the relation $=_q$ into a single variable. This transformation gives us a *canonical* renaming from $\mathit{vars}(q)$ to $\mathit{vars}(q^{\approx})$. For instance, $q \mathrel{\hat{=}} \exists x, y, z \;\; x \xrightarrow{K} y \wedge y \xrightarrow{L} z \wedge (x = y)$ collapses to $q^{\approx} \mathrel{\hat{=}} \exists x, z \;\; x \xrightarrow{K} x \wedge x \xrightarrow{L} z$.

For any atom $x \xrightarrow{L} y$ of a CRPQ q and $w = a_1 a_2 \ldots a_k \in L$, the *w-expansion* of $x \xrightarrow{L} y$ is the CQ P defined as

follows: (i) If $w \neq \varepsilon$ then $P \doteq x \xrightarrow{a_1} z_1 \wedge z_1 \xrightarrow{a_2} z_2 \wedge \cdots \wedge z_{k-1} \xrightarrow{a_k} y$ where each z_i is a fresh variable (ii) If $w = \varepsilon$ then $P \doteq (x = y)$. By a slight abuse of notation, we write $P \doteq x \xrightarrow{w} y$ to denote such a w-expansion, with $w = a_1 \cdots a_k$ where the z_i variables are implicit. For $m \in \mathbb{N}$, an *atom m-expansion* of $x \xrightarrow{L} y$ is a w-expansion for some $w \in L$ such that $|w| \leq m$. An *m-expansion* of a CRPQ q is the CQ resulting from (i) replacing each atom with an atom m-expansion and (ii) removing the equality atoms canonically. An *atom expansion* is an atom m-expansion for some m, likewise an *expansion* is an m-expansion for some m. For instance, for the atom $A = (x \xrightarrow{ab^*c} y)$, one example of a 3-expansion would be $x \xrightarrow{a} z_1 \xrightarrow{b} z_2 \xrightarrow{c} y$. A does not have any 1-expansion.

Let $\mathrm{Exp}(q)$ denote the (possibly infinite) set of all expansions of q. and let $\mathrm{Exp}_m(q) \subseteq \mathrm{Exp}(q)$ denote the (finite) set of all m-expansions of q. A *homomorphism* from a (Boolean) CRPQ q to a graph database $G = (V, E)$, is defined to be any homomorphism $h : \mathit{vars}(\lambda) \xrightarrow{hom} G$ for some $\lambda \in \mathrm{Exp}(q)$. We further say that G *satisfies* q, denoted by $G \models q$, if there is such a homomorphism. We will consider $\mathrm{Exp}_m(q)$ as a query, which holds true in a graph database G if $G \models \lambda$ for some $\lambda \in \mathrm{Exp}_m(q)$.

A *union of CRPQs* (*UCRPQ*) is of the form $q \doteq \bigvee_{1 \leq i \leq n} q_i$, where each q_i is a CRPQ. The set of expansions of a UCRPQ is the union of the expansions of the CRPQs therein. Similarly, *Unions of CQs* (*UCQs*) are finite unions of CQs. As mentioned earlier, we assume that UCRPQs/UCQs are Boolean, in the sense that they only contain Boolean CRPQs/CQs. A graph database satisfies a union of CRPQ (resp. CQs) if it satisfies at least one of its disjuncts.

For any syntactic object $\mathcal{O}$, we write $\mathit{vars}(\mathcal{O})$ to denote the set of variables it contains. Let $|q|_{\mathsf{at}}$ and $|q|_{\mathsf{var}}$ be the number of atoms and variables of q, respectively.

Given two Boolean queries q and q' (which may be (U)CQs, (U)CRPQs, $\mathrm{Exp}_m(q)$-expansions, etc.), we write $q \subseteq q'$ if for every graph database G such that $G \models q$ we have $G \models q'$, in which case we say that q is *contained* in q'. We say that q and q' are *equivalent*, written $q \equiv q'$, if $q \subseteq q'$ and $q' \subseteq q$. The following lemma characterizes the containment in terms of expansions and homomorphisms.

Lemma 1 (Folklore)**.** *Given two UCRPQs q and q', we have $q \subseteq q'$ if, and only if, for every $\lambda \in \mathrm{Exp}(q)$ there is $\lambda' \in \mathrm{Exp}(q')$ such that there exists a homomorphism $\lambda' \xrightarrow{hom} \lambda$.*

A UCRPQ q is *bounded* if it is equivalent to some UCQ Φ. The following key property characterizes boundedness in terms of m-expansions.

Proposition 2. *(Barceló, Figueira, and Romero 2019, Proposition 3) A UCRPQ q is bounded if, and only if, q is equivalent to $\mathrm{Exp}_m(q)$ for some $m \in \mathbb{N}$.*

The BOUNDEDNESS problem for a class $\mathcal{C} \subseteq$ UCRPQ of queries is the problem of, given a query $q \in \mathcal{C}$ whether q is bounded. This is the main problem we will study in this paper. While the BOUNDEDNESS problem has been shown to be decidable for UCRPQ (as stated below), we will focus on small fragments thereof.

Theorem 3. *(Barceló, Figueira, and Romero 2019, Theorem 11)* BOUNDEDNESS *for UCRPQs is* ExpSpace*-complete. Further, the upper-bound holds also in the presence of two-way navigation, and the lower bound holds already for CRPQs.*

3 Main Results

The lower bound in Theorem 3 uses regular expressions that are rather complex. However, as explained in Section 1, queries used in practice often contain simple regular expressions. Our results focus on UCRPQs where atoms are of the form a^*, where $a \in \mathbb{A}$ is a label, and more generally of the form w^*, where $w \in \mathbb{A}^*$. We next formally define this restricted class of regular expressions.

Let $\mathrm{RE}(\mathbb{A})$ denote the set of all regular expressions over $\mathbb{A}$ using concatenation ($\square \cdot \square$), union ($\square + \square$), Kleene star ($\square^*$) from the basic letter-expressions (a, for each $a \in \mathbb{A}$) and from the symbol ε denoting the empty string. Let $\mathbf{a} \subseteq \mathrm{RE}(\mathbb{A})$ denote regular expressions of the form $a \in \mathbb{A}$ (atomic regular expressions) and $\mathbf{a}^*$ denote regular expressions of the form a^* where $a \in \mathbb{A}$. We write $\mathbf{a}^*_A \subseteq \mathrm{RE}(\mathbb{A})$ for some $A \subseteq \mathbb{A}$ to denote the regular expressions of the form a^* where $a \in A$. Similarly, let $\mathbf{w} \subseteq \mathrm{RE}(\mathbb{A})$ denote regular expressions of the form $w \in \mathbb{A}^*$ (*i.e.*, expressions of the form $w = a_1 \cdots a_n$ with $a_i \in \mathbb{A}$) and $\mathbf{w}^*$ denote regular expressions of the form w^* where $w \in \mathbb{A}^*$.

Let SF (star-free expressions) denote the set of all *star-free regular expressions* over $\mathbb{A}$, that is, all expressions formed using $\{\varepsilon, (a)_{a \in \mathbb{A}}, +, \cdot\}$. We will further consider the class SSF of *succinct star-free expressions* which are star-free regular expressions which additionally allow for expressions of the form w^n and $w^{\leq n}$, where $w \in \mathbb{A}^*$ and n is encoded in binary. The expressions w^n and $w^{\leq n}$ are succinct representations of $\underbrace{w \cdots w}_{n \text{ times}}$ and $\underbrace{(\varepsilon + w) \cdots (\varepsilon + w)}_{n \text{ times}}$.

Observation 4. *Observe that $\mathbf{a} \subsetneq$ SF $\subsetneq$ SSF and $\mathbf{a}^* \subsetneq \mathbf{w}^*$. However,* SF *and* SSF *are equivalent in terms of expressive power.*

The *size* of an SSF expression is the number of symbols needed for its encoding, for example the size of w^n is $|w| + \lceil \log(n) \rceil$. The size of an expression w^* is $|w|$.

The *size* of an atom is the size of the regular expression therein. The *size* of a CRPQ or CQ is the sum of sizes of all its atoms and the *size* of a UCRPQ or UCQ is the sum of the sizes of all its CRPQs or CQs.

For any given class C of regular expressions (denoting regular languages), let UCRPQ(C) be the class of UCRPQs whose atoms contain languages specified by expressions from C. We write UCRPQ(C_1, C_2) to denote UCRPQ($C_1 \cup C_2$) and CRPQ(C) to denote the subclass of UCRPQ(C) consisting of CRPQs. For instance, a query in CRPQ(SSF, $\mathbf{w}^*$) can have an edge label of the form $(ab)^n$ (where n is written in binary) or $(abb)^*$.

The goal of the paper is to prove that BOUNDEDNESS for UCRPQ(SSF, $\mathbf{w}^*$) is Π^p_2-complete and, further, that the bounded query can be produced efficiently.

Theorem 5. BOUNDEDNESS *for* UCRPQ(SSF, $\mathbf{w}^*$) *is* Π^p_2*-complete. The problem remains* Π^p_2*-hard for* CRPQ($\mathbf{a}, \mathbf{a}^*$)*. Moreover, if* $q \in$ UCRPQ(SSF, $\mathbf{w}^*$) *is bounded then it is equivalent to a query* $q' \in$ UCRPQ(SSF) *of linear size which can be computed.*

For proving Theorem 5, we will use the membership problem for non-deterministic finite automata (NFA) whose transitions may be succinctly represented as $q \xrightarrow{w^n} q'$, where n is encoded in binary, which we call succinct-NFA. We will show in Section 4 that the membership problem for such succinctly represented automata is still in NP, which is a result of independent interest.

Theorem 6. *The membership problem for succinct-NFA is in* NP*.*

If a query q in UCRPQ(SSF, $\mathbf{a}^*$) is not bounded, then it is natural to ask if it can be bounded "as much as possible". Intuitively, we want a query q' that is equivalent to q where the atoms of q' are either SSF expressions or a^* only for those letters $a \in \mathbb{A}$ that cannot be bounded in q. Formally, for any given $A \subseteq \mathbb{A}$, we say q is A-bounded if it is equivalent to a query from UCRPQ(SSF, $\mathbf{a}^*_{\bar{A}}$) for $\bar{A} = \mathbb{A} \setminus A$. Observe that such q is bounded *iff* it is $\mathbb{A}$-bounded. The problem of checking whether $q \in$ UCRPQ(SSF, $\mathbf{a}^*$) is A-bounded is called the A-BOUNDEDNESS problem.

Theorem 7. *For the class* UCRPQ(SSF, $\mathbf{a}^*$) *of queries:*

1. *The* A-BOUNDEDNESS *problem is* Π^p_2*-complete. The problem remains* Π^p_2*-hard even for* CRPQ($\mathbf{a}, \mathbf{a}^*$)*. Moreover, an equivalent query* $q' \in$ UCRPQ(SSF, $\mathbf{a}^*_{\bar{A}}$) *of linear size can be computed.*
2. *There is a unique maximal* $A \subseteq \mathbb{A}$ *such that* q *is* A*-bounded, and there is a* Π^p_2 *algorithm for finding it.*

4 Succinct Automata and Succinct CQs

In this section we study a problem, of independent interest, which will be necessary to obtain our upper bounds (more precisely, the upper bound of Theorem 5, covered by Theorem 11).

Let us define a succinct-NFA over an alphabet $\mathbb{A}$ as a classical non-deterministic finite automaton (*NFA*) where transitions can be of the form $q \xrightarrow{w^n} q'$ where $w \in \mathbb{A}^*$ and $n \in \mathbb{N}$ is encoded in binary. More formally, a *succinct-NFA* over $\mathbb{A}$ is a tuple (Q, δ, q_0, F), where $q_0 \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *final states*, and $\delta \subseteq_{fin} Q \times \mathbb{A}^* \times \mathbb{N} \times Q$ is a finite set of *succinct-transitions*. We sometimes write $q \xrightarrow{w^n} p$ instead of (q, w, n, p), and n is always encoded in binary. An *accepting run* of a given succinct-NFA $\mathcal{A}$ is, as expected, a sequence of transitions $(p_0, w_1, n_1, p_1), (p_1, w_2, n_2, p_2), \ldots, (p_{m-1}, w_m, n_m, p_m)$ such that p_0 is initial state, and p_m is a final state. The word from $\mathbb{A}^*$ associated to the accepting run is $w_1^{n_1} \cdots w_m^{n_m} \in \mathbb{A}^*$. The language associated to a given succinct-NFA $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set of all words associated to accepting runs.

Given a word $w = a_1 \cdots a_n \in \mathbb{A}^*$ over a finite alphabet $\mathbb{A}$, we define the factor of w between i and j, denoted by $w[i..j]$, as ε if $j \leq i$ or $a_{i+1} \cdots a_j$ otherwise.

The *membership problem* for succinct-NFA is the problem of, given a word $v \in \mathbb{A}^*$, a number $m \in \mathbb{N}$ (in binary), and a succinct-NFA $\mathcal{A}$, whether $v^m \in \mathcal{L}(\mathcal{A})$. We observe that this is somewhat close to the automata on compressed strings of (Martens, Neven, and Schwentick 2009, §2.3), although in their case, the succinct representation is not on the NFA but only on the input word for testing membership.

Theorem 6 (Restatement)**.** *The membership problem for succinct-NFA is in* NP*.*

Proof. Let $\mathcal{A}$ be a succinct-NFA, $v \in \mathbb{A}^*$ be a word and m be a number represented in binary. We assume *w.l.o.g.* that all numbers in δ are positive (*i.e.* there are no ε transitions).

We first build a new succinct-NFA $\mathcal{A}'$ from $\mathcal{A}$ so that $v^m \in \mathcal{L}(\mathcal{A})$ *iff* $\mathcal{L}(\mathcal{A}')$ contains a word of length $m \times |v|$. We define it as follows. The set of states of $\mathcal{A}'$ is $Q \times \{0, \ldots, |v| - 1\}$.

There is a succinct-transition $(q, i) \xrightarrow{w^n} (p, j)$ if (i) $q \xrightarrow{w^n} p$ is a succinct-transition of $\mathcal{A}$, and (ii) w^n is of the form $v[i..|v|] \cdot v^\ell \cdot v[0..j]$ for some $\ell \geq 0$, or equal to $v[i..j]$. The initial state of $\mathcal{A}'$ is $(q_0, 0)$ and the set of final states is $F \times \{0\}$.

Claim 8. *We can build* $\mathcal{A}'$ *in polynomial time.*

Proof. If suffices to prove that checking whether w^n is of the form $v[i..|v|] \cdot v^\ell \cdot v[0..j]$ for some $\ell \geq 0$, or equal to $v[i..j]$ is indeed in polynomial time. Towards this first we build a directed graph G having $\{0, \ldots, |v|\}$ as vertices, and an edge $i \to j$ if $w = v[i..|v|] \cdot v^\ell \cdot v[0..j]$ for some ℓ. Notice that ℓ is bounded by $|w|$ and hence G can be built in polynomial time.

Now in order to check if w^n is of the form $v[i..|v|] \cdot v^\ell \cdot v[0..j]$ for some $\ell \geq 0$, it suffices to check if there is a path of length n from i to j in G. The set of all sizes of paths from i to j in G can be seen as the Parikh-image[2] of an NFA over a one-letter alphabet. Since the problem of testing membership of a vector in the Parikh-image of an NFA is in polynomial time as soon as the alphabet is bounded (Kopczyński and To 2010), the statement follows. □

Claim 9. $v^m \in \mathcal{L}(\mathcal{A})$ *iff* $\mathcal{L}(\mathcal{A}')$ *contains a word of length* $m \times |v|$*.*

Proof. Left-to-right implication. Assuming $v^m \in \mathcal{L}(\mathcal{A})$ consider an accepting run $(p_0, w_1, n_1, p_1), (p_1, w_2, n_2, p_2), \ldots, (p_{t-1}, w_t, n_t, p_t)$ on v^m. Then, for each $i \in \{1, \ldots, t\}$, we have that $w_1^{n_1} \cdots w_i^{n_i}$ is a prefix of v^m and thus can be written as $v^{m_i} \cdot v[0..j_i]$ for some j_i and m_i, where in particular $j_t = 0$ and $m_t = m$.

Hence, $((p_0, 0), w_1, n_1, (p_1, j_1)), ((p_1, j_1), w_2, n_2, (p_2, j_2)), \ldots, ((p_{t-1}, j_{t-1}), w_t, n_t, (p_t, j_t))$ is an accepting run of $\mathcal{A}'$ on v^m.

Right-to-left implication. First observe that $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$ since an accepting run of $\mathcal{A}'$ on a word w induces an accepting run of $\mathcal{A}$ on the same word by projecting the run onto Q.

[2] Remember that the *Parikh-image* of a language $L \subseteq \mathbb{A}^*$ over a one-letter alphabet $\mathbb{A} = \{a\}$ is $\{t \in \mathbb{N} : a^t \in L\}$.

Observe that an accepting run of $\mathcal{A}'$ on a word w forces w to be of the form v^ℓ. Since $\mathcal{L}(\mathcal{A}')$ has an accepted word of length $m \times |v|$, this forces $\ell = m$, *i.e.*, the accepted word has to be exactly v^m. Since $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$ by projecting the run onto Q, we obtain $v^m \in \mathcal{L}(\mathcal{A})$. □

Claim 10. *Testing if $\mathcal{L}(\mathcal{A}')$ contains a word of a given length is in* NP.

Proof. By replacing each transition $q \xrightarrow{w^n} p$ in $\mathcal{A}'$ with $q \xrightarrow{|w|\times n} p$ we obtain a "succinct one counter automaton", whose reachability problem is in NP (Haase et al. 2009, Theorem 1). □

This concludes the proof of Theorem 6. □

4.1 Containment Problem for Succinct CQs

A *succinct CQ* is a CRPQ(SSF) whose every atom has an expression of the form w^n for $w \in \mathbb{A}^*$, $n \in \mathbb{N}$, with the expected semantics. Remember that we assume that n is given in binary and that the size of each atom w^n is $|w| + \lceil \log(n) \rceil$, and hence that every succinct CQ has an equivalent *corresponding* CQ of at most exponential size (*i.e.*, its only expansion). The *containment problem* for a class $\mathcal{Q}$ of queries is the problem of, given two queries $q, q' \in \mathcal{Q}$, whether $q \subseteq q'$.

Theorem 11. *The containment problem for succinct CQs is in* NP.

Proof. Given two succinct CQs q, q' it suffices to check if there is a homomorphism from $\tilde{q}'$ to $\tilde{q}$, where $\tilde{q}'$ and $\tilde{q}$ are the CQs corresponding to q' and q, respectively. Towards this, we first non-deterministically "break" some atoms $x \xrightarrow{w^n} x'$ of q introducing new variables $x \xrightarrow{w_1^{n_1}} y_1 \wedge y_1 \xrightarrow{w_2^{n_2}} y_2 \wedge \cdots \wedge y_{k-1} \xrightarrow{w_k^{n_k}} y_k$ where $w^n = w_1^{n_1} \cdots w_k^{n_k}$ in such a way that we introduce at most $|vars(q')|$ new variables. To verify that $w^n = w_1^{n_1} \cdots w_k^{n_k}$, we can build a succinct-NFA of the form $p_0 \xrightarrow{w_1^{n_1}} p_1 \ldots p_{k-1} \xrightarrow{w_k^{n_k}} p_k$ with p_0 and $\{p_k\}$ as initial and final states respectively and check if w^n is accepted by this automaton (which can be done in NP, by Theorem 6).

This results in a succinct CQ $\hat{q}$ which is equivalent to q (in fact, $\tilde{q}$ is still its corresponding CQ) and of polynomial —even linear— size. We now guess a function f from the variables of q' to the variables of $\hat{q}$. Finally, for each atom $x \xrightarrow{w^n} y$ of q', we check if there is a path from $f(x)$ to $f(y)$ in $\hat{q}$ reading w^n. For this, we can see $\hat{q}$ as a succinct-NFA whose initial state is $f(x)$ and its set of final states is $\{f(y)\}$, and where we check if w^n belongs to its language, which is in NP by Theorem 6. □

5 Upper Bound

In this section we prove the upper bounds of Theorem 5. Formally, we will prove the following statement.

Theorem 12. *The* BOUNDEDNESS *problem for* UCRPQ(SSF, $\mathbf{w}^*$) *is in* Π_2^p. *Further, an equivalent* UCRPQ(SSF) *query of linear size can be computed.*

The goal of this section is to prove Theorem 12. From Observation 4, the bound applies also to UCRPQ($\mathbf{w}, \mathbf{w}^*$) $\subseteq$ UCRPQ(SF, $\mathbf{w}^*$) $\subseteq$ UCRPQ(SSF, $\mathbf{w}^*$).

For $q \in$ UCRPQ(SSF, $\mathbf{w}^*$) and $m \in \mathbb{N}$, let $q(m) \in$ UCRPQ(SSF) be the result of replacing each regular expression of the form w^* in q with $w^{\leq m}$. We begin with the following observation.

Observation 13. *If $q \equiv \mathrm{Exp}_m(q)$, then $q \equiv q(m)$. If $q \equiv q(m)$ then $q \equiv \mathrm{Exp}_{m \cdot m'}(q)$ for m' the maximum length of a word w in an expression of the form w^*. Hence, by Proposition 2, $q \in$ UCRPQ(SSF, $\mathbf{w}^*$) is bounded if, and only if, q is equivalent to $q(\ell)$ for some $\ell \in \mathbb{N}$.*

In view of the previous observation, we will rather work with the more intuitive query $q(m)$ rather than $\mathrm{Exp}_m(q)$. For economy of space we will simply write "$\lambda \in q(m)$" to denote $\lambda \in \mathrm{Exp}(q(m))$.

Recursive and non-recursive atoms. An atom of the form $x \xrightarrow{w^*} y$ is called *recursive*. Let $\mathcal{R}_q \subseteq \mathbb{A}^*$ be the set of all words w from the labels w^* of recursive atoms of q. Let $N_q \in \mathbb{N}$ be the maximum length of a word in a non-recursive atom of q. We will prove Theorem 12 via the following intermediate results.

Lemma 14. *If $q \in$ UCRPQ(SSF, $\mathbf{w}^*$) is bounded, then it is equivalent to $q(Z_q)$ for $Z_q \doteq |q|_{\mathsf{at}}^3 \cdot N_q \cdot |q|_{\mathsf{var}} \cdot \Pi_{w \in \mathcal{R}_q} |w|$.*

Lemma 15. *A query $q \in$ UCRPQ(SSF, $\mathbf{w}^*$) is bounded iff $q(Z_q)$ is equivalent to $q(Z_q^+)$, for $Z_q^+ \doteq |q|_{\mathsf{at}} \cdot Z_q + 1$.*

Lemma 15 reduces the BOUNDEDNESS problem for UCRPQ(SSF, $\mathbf{w}^*$) to the containment problem for succinct CQs. The proof of Lemma 15 will use Lemma 14. Assuming these two results, we will first prove Theorem 12.

Proof of Theorem 12. From Lemma 15, to test whether a query q is bounded can be reduced to checking whether $q(n) \equiv q(n')$ for some $n, n' \in \mathbb{N}$ of polynomial space (in binary). Given that the "succinct" labels of SSF expressions appearing in q have the form $w^{\leq m}$ or w^m with m in binary, the expansions of $q(n)$ and $q(n')$ are at most single-exponential in the size of q. Moreover, every expansion of $q(n)$ can be expressed as a succinct CQ of polynomial size.

In order to check that $q(n) \subseteq q(n')$ does not hold, we first guess an n-expansion λ of q that is not contained in $q(n')$, as a succinct CQ of polynomial size. This is possible by the discussion above. Then we check that $\lambda \subseteq q(n')$ does not hold. To check $\lambda \subseteq q(n')$, it suffices to guess an n'-expansion λ' of q, which, once again, we assume is a succinct CQ of polynomial size, and then check that $\lambda \subseteq \lambda'$. This is in NP by Theorem 11; thus, we have a procedure which is in co-NP for testing $\lambda \not\subseteq \lambda'$.

This gives a Σ_2^p algorithm to test $q(n) \not\subseteq q(n')$, in other words a Π_2^p algorithm to test $q(n) \subseteq q(n')$, and thus whether q is bounded.

Finally, the existence of the equivalent query is trivial since $q(Z_q)$ can be produced in linear time. □

Hence it is enough to prove Lemma 15 which in turn needs Lemma 14. The next two subsections are focused on the proofs of these two lemmas.

5.1 Proof of Lemma 14

We recall the statement of Lemma 14:

Lemma 14. *If $q \in$ UCRPQ(SSF, $\mathbf{w}^*$) is bounded, then it is equivalent to $q(Z_q)$.*

Before proving the result formally, we describe the key ideas informally. Suppose q is bounded, then by Observation 13, it is equivalent to $q(M)$ for some M. Assume for contradiction that this M is necessarily larger than Z_q. This means that there exists some $\lambda \in \text{Exp}(q)$ such that for every $\hat{\lambda} \in q(Z_q)$ there is no homomorphism from $\hat{\lambda}$ to λ but there is some $\lambda' \in q(M)$ (of minimal size) such that there is a homomorphism $h : \lambda' \xrightarrow{hom} \lambda$. Since $\lambda' \notin q(Z_q)$ there is some atom $x \xrightarrow{w^*} y$ in q such that it is expanded as $x \xrightarrow{w^l} y$ in λ' for some $l > Z_q$. The choice of Z_q is large enough such that we can argue that using the homomorphism $h : \lambda' \xrightarrow{hom} \lambda$, we can transform λ' into some λ'' (by contracting some recursive expansions) such that there is a homomorphism from λ'' to λ. This would contradict the minimality of λ'. The main technical difficulty is then to produce λ'' and the homomorphism from λ'' to λ.

To prove Lemma 14 we shall need the following claim.

Claim 16. *There is $\lambda' \in q(M)$ and $h : \lambda' \xrightarrow{hom} \lambda$ such that the size of the image Im(h) of h is at most Z_q.*

We first prove that Claim 16 implies Lemma 14.

Proof of Lemma 14. By means of contradiction, suppose q is bounded but not equivalent to $q(Z_q)$. Then, there is no $\hat{\lambda} \in q(Z_q)$ such that there is a homomorphism $\hat{\lambda} \xrightarrow{hom} \lambda$. By Claim 16, there is some $\lambda' \in q(M)$ and $h : \lambda' \xrightarrow{hom} \lambda$ such that the size of the image *Im*(h) of h is at most Z_q. Pick a minimal[3] λ' that satisfies this property.

By assumption, $\lambda' \notin q(Z_q)$. Hence, there is some recursive atom expansion $x \xrightarrow{w^\ell} y = x \xrightarrow{w} x_1 \xrightarrow{w} \cdots \xrightarrow{w} x_{\ell-1} \xrightarrow{w} y$ of λ' where $\ell > Z_q$.[4] By Claim 16 and the pigeonhole principle, there will be two variables x_i, x_j such that $h(x_i) = h(x_j)$ for some $i < j$. But then we can replace the atom expansion with $x \xrightarrow{w^{\ell-(j-i)}} y = x \xrightarrow{w} x_1 \xrightarrow{w} \cdots \xrightarrow{w} x_i \xrightarrow{w} x_{j+1} \xrightarrow{w} \cdots \xrightarrow{w} x_{\ell-1} \xrightarrow{w} y$, obtaining an expansion $\lambda'' \in q(M)$ which is strictly smaller than λ' such that h restricted to λ'' forms a homomorphism $h' : \lambda'' \xrightarrow{hom} \lambda$. Moreover, since $h' \subseteq h$, the size of *Im*(h') is still at most Z_q, contradicting the minimality of λ'. □

Thus it remains to prove Claim 16.

Proof of Claim 16. Let M' be the size of $q(M)$ (seen as a UCQ). Let λ^+ be the result of replacing each atom expansion $x \xrightarrow{w^\ell} y$ of λ such that $\ell > Z_q$ with $x \xrightarrow{w^{M'}} y$. Notice that $\lambda^+ \in \text{Exp}(q)$ —indeed, atom expansions of this kind can only come from recursive atoms (because Z_q is sufficiently large). Now pick a $\lambda' \in q(M)$ such that λ' is of minimal size and there is a homomorphism $h : \lambda' \xrightarrow{hom} \lambda^+$. We show the following:

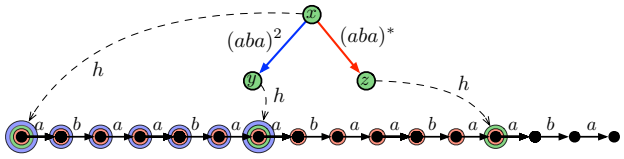

Figure 1: Consider the query $x \xrightarrow{(aba)^n} y \wedge x \xrightarrow{(aba)^*} z$, where $(aba)^n$ is a SSF with $n = 11$ in binary (*i.e.*, 3).. Below, we have the corresponding coloring scheme in λ^+ with the respective colors according to a homomorphism h.

(1) $|Im(h)| \leq Z_q$;

(2) h is also a homomorphism to (an isomorphic copy[5] of) λ, *i.e.*, $h : \lambda' \xrightarrow{hom} \lambda$.

(1) Let us start with the expansion $\lambda' \in q(M)$ such that $h : \lambda' \xrightarrow{hom} \lambda^+$. First, we devise a color scheme for the variables of λ^+.

Let us color with *blue* any variable of λ^+ which is the h-image of a variable of a non-recursive atom expansion of λ'. Color a variable *green* if it is the h-image of a variable of q, and color it *red* if it is the h-image of a variable from a recursive atom expansion. Of course, some variables may be colored with multiple colors (see Figure 1).

Consider any recursive atom expansion of λ^+ of the form $x \xrightarrow{w^{M'}} y$. Recall that M' is the size of the M-expansion $q(M)$ considered as a UCQ, and $\lambda' \in q(M)$. Since M' is larger than any expansion of $q(M)$ (in particular λ'), the expansion $x \xrightarrow{w^{M'}} y$ must contain some uncolored variables.

Let $x_0, \ldots, x_{M'}$ be the variables of the path $x \xrightarrow{w^{M'}} y$, in order, with $x_0 = x$ and $x_{M'} = y$. If there is an *interval* $I = \{x_i, x_{i+1}, \ldots, x_j\}$ with $j - i = Z_{red} \hat{=} \Pi_{w \in \mathcal{R}_q} |w|$ which is *purely-red* (*i.e.*, whose every variable is colored red and by no other color), this means that the h-preimage of I consists of intervals of recursive atom expansions of the form $\tilde{w}^k$ for some $\tilde{w}$ and k such that $|\tilde{w}^k| = Z_{red}$ (note that $\tilde{w}$ need not be a recursive atom, it could be some word in $\mathbb{A}^*$). Concretely, $h^{-1}(I) = I_1 \cup \cdots \cup I_r$ such that each I_s is of the form $I_s = \{z_{(s,0)}, \ldots, z_{(s,Z_{red})}\}$ where $h(z_{(s,t)}) = x_{i+t}$ for every s, t and I_s forms a directed path reading some $\tilde{w}^k$.

Consider λ'' as the result of *contracting* these intervals I_s from λ': for each interval I_s, we remove all atoms of λ' inside I_s, and we rename $z_{(s,0)}$ as $z_{(s,Z_{red})}$(see Figure 2). Since this operation corresponds to removing some iterations of a

[3] By minimal λ' we mean in terms of the number of variables.

[4] Remember that a recursive atom expansion of λ' is simply an expansion of a recursive atom of q in λ'.

[5] By *isomorphic copy* of λ we mean an expansion $\tilde{\lambda}$ such that there is a bijection f between the variables of λ and $\tilde{\lambda}$ such that $x \xrightarrow{a} y$ is an atom of λ *iff* $f(x) \xrightarrow{a} f(y)$ is an atom of $\tilde{\lambda}$.

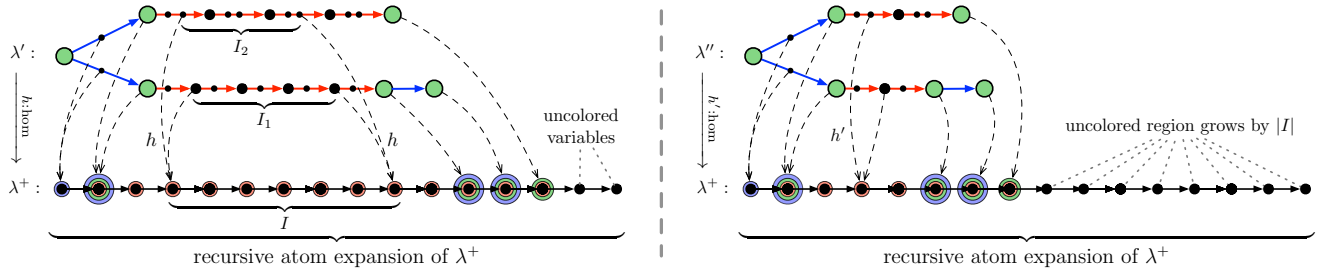

Figure 2: (Left) An example of a homomorphism from λ' to λ^+ with a large purely-red interval I. In λ', red (resp. blue) edges correspond to expansions of recursive atoms (resp. non-recursive atoms), and green vertices correspond to variables of q. The h-preimage of I contains two intervals I_1 and I_2 coming from recursive atom expansions of words of length 2 and 3 respectively.
(Right) The resulting λ''. The intervals I_1, I_2 are contracted to a single variable in the homomorphism to λ^+. The image of the variables which appeared to the right of I are now shifted to the left, making the number of uncolored variables grow by $|I| - 1$.

recursive atom expansion (and *only* recursive atom expansion since the color is *exclusively* red), we have $\lambda'' \in q(M)$.

Further, note that since there is always at least one uncolored variable λ^+ (which is a recursive atom expansion), we have $\lambda'' \xrightarrow{hom} \lambda^+$ because it suffices to shift the image of the variables appearing of one side of I as shown in Figure 2, in particular making the uncolored region grow. This is in contradiction with λ' being minimal.

Hence, in an atom expansion $x \xrightarrow{w^{M'}} y$ of λ^+, there cannot be a purely-red interval of size Z_{red}. On the other hand, since $|q|_{\mathsf{var}}$ is the number of variables in q, there cannot be more than $|q|_{\mathsf{var}}$ green-colored variables. Finally, there cannot be more than $|q|_{\mathsf{at}} \cdot N_q$ blue-colored variables.

Thus, there cannot be a colored interval of size greater than $Z_{col} \mathrel{\hat{=}} |q|_{\mathsf{at}} \cdot N_q \cdot |q|_{\mathsf{var}} \cdot Z_{red}$ (as it would imply a Z_{red}-sized purely-red interval). Since $|q|_{\mathsf{at}}$ is the number of atoms, there cannot be more than $|q|_{\mathsf{at}}$ colored intervals of maximal size in such atom expansions $x \xrightarrow{w^{M'}} y$ (actually, the number of connected components of q would suffice).

How many colored variables does λ^+ have? Since the maximum length of a colored interval is Z_{col} and since there are no more than $|q|_{\mathsf{at}}$ intervals on any given atom expansion of λ^+, there cannot be more than $Z_q = |q|_{\mathsf{at}}^2 \cdot Z_{col}$ colored variables in λ^+. Hence, $|Im(h)| \leq Z_q$.

(2) Next, we show that h is also a homomorphism to an isomorphic copy of λ. Observe that if we have an uncolored interval of $x \xrightarrow{w^{M'}} y$ of size $|w|$ we can simply contract it obtaining some $\lambda^{\pm}$ such that h is still a homomorphism $\lambda' \xrightarrow{hom} \lambda^{\pm}$ and $\lambda^{\pm} \in \mathrm{Exp}(q)$.

Consider any recursive atom expansion from λ^+ of the form $x \xrightarrow{w^{M'}} y$. By construction of λ^+, recall that $x \xrightarrow{w^{M'}} y$ was obtained by replacing some atom expansion $x \xrightarrow{w^m} y$ of λ where $m > Z_q$. If we could find $M' - m$ such uncolored intervals of an atom expansion $x \xrightarrow{w^{M'}} y$ as before, we could contract them and obtain (an isomorphic copy of) $x \xrightarrow{w^m} y$. Further, h would still be a valid homomorphism.

But how many such uncolored $|w|$-sized intervals are there? The worst-case scenario in which we could not find any such uncolored interval is the one where every pair of colored-intervals (and there are at most $|q|_{\mathsf{at}}$ many) separated by $|w| - 1$ uncolored-intervals. Since the size and number of colored-intervals is bounded by Z_{col} and $|q|_{\mathsf{at}}$ respectively, the length of such worst-case atom expansion is bounded by $Z_{worst} \mathrel{\hat{=}} |q|_{\mathsf{at}} \cdot Z_{col} + (|q|_{\mathsf{at}} + 1) \cdot (|w| - 1)$. Hence, there are at least $\lceil ((M' \cdot |w|) - Z_{worst})/|w| \rceil$ uncolored-intervals of size $|w|$, and since

$$\begin{aligned}
&\lceil ((M' \cdot |w|) - Z_{worst})/|w| \rceil \\
&\geq M' - Z_{worst}/|w| \\
&\geq M' - |q|_{\mathsf{at}} \cdot Z_{col} \cdot |w| \\
&\geq M' - Z_q \qquad \text{(since } Z_q > |q|_{\mathsf{at}} \cdot Z_{col} \cdot |w|\text{)} \\
&\geq M' - m, \qquad \text{(since } m > Z_q\text{)}
\end{aligned}$$

we can contract $(M' - m)$-many uncolored intervals among the available ones. If we repeat these contractions for every M'-atom expansion, we obtain (an isomorphic copy of) λ from λ^+, which shows $h : \lambda' \xrightarrow{hom} \lambda$. □

This concludes the proof of Claim 16 and hence of Lemma 14.

5.2 Proof of Lemma 15

We will actually prove the following statement, which entails Lemma 15.

Lemma 17. *For every query* $q \in$ UCRPQ(SSF, $\mathbf{w}^*$)*, the following are equivalent:*

(1) q *is bounded,*
(2) $q \equiv q(Z_q)$*,*
(3) $q(Z_q^+) \equiv q(Z_q)$*.*

Proof. (2) ⇒ (1) is by definition of being bounded.

(1) ⇒ (3) By the previous Lemma 14, if q is bounded then it is equivalent to $q(Z_q)$, and hence also to $q(Z_q^+)$ since $Z_q^+ \geq Z_q$.

(3) ⇒ (2) We show the contrapositive statement, namely that $q \not\subseteq q(Z_q)$ implies $q(Z_q^+) \not\subseteq q(Z_q)$. Assume that $q \not\subseteq$

$q(Z_q)$. Then there is some expansion $\lambda \in \text{Exp}(q)$ such that $\lambda \not\subseteq q(Z_q)$. Observe that the size of every expansion from $q(Z_q)$ is strictly bounded by $Z_q^+ = |q|_{\mathsf{at}} \cdot Z_q + 1$. Consider the expansion λ' as the result of replacing each atom expansion $x \xrightarrow{w^\ell} y$ of λ such that $\ell > Z_q$ with $x \xrightarrow{w^{Z_q^+}} y$. Clearly, $\lambda' \in q(Z_q^+)$.

We want to show that $\lambda' \not\subseteq q(Z_q)$. By means of contradiction, assume that there is an expansion $\hat{\lambda} \in q(Z_q)$ such that $h : \hat{\lambda} \xrightarrow{hom} \lambda'$. Then, each replaced atom expansion $x \xrightarrow{w^{Z_q^+}} y$ of λ' must contain at least one vertex which is not in the image of h. Then, we can expand it back[6] to $x \xrightarrow{w^\ell} y$ still results in a homomorphism. Therefore, $\hat{\lambda} \xrightarrow{hom} \lambda'$ implies $\hat{\lambda} \xrightarrow{hom} \lambda$; and since $\lambda \not\subseteq q(Z_q)$, we have $\lambda' \not\subseteq q(Z_q)$.

Hence, $q \not\subseteq q(Z_q)$ implies $q(Z_q^+) \not\subseteq q(Z_q)$. □

6 Lower Bound

Here we show that even for the simplest fragment, namely CRPQ where the regular expressions are of the form a^* or just a, the BOUNDEDNESS problem is already Π_2^p-hard.

Theorem 18. *The* BOUNDEDNESS *problem for CRPQ$(\mathbf{a}, \mathbf{a}^*)$ is Π_2^p-hard.*

The proof goes by reduction from ∀∃-QBF satisfiability. Consider an instance of ∀∃-QBF

$$\Phi = \forall x_1, \ldots, x_n \exists y_1, \ldots, y_l \; \varphi(x_1, \ldots, x_n, y_1, \ldots, y_l) \tag{1}$$

where φ is quantifier free and in 3-CNF. We define Boolean CRPQ$(\mathbf{a}, \mathbf{a}^*)$ queries q_1 and q_2 on the alphabet $\mathbb{A} = \{b, a, s, j, x_1, \ldots, x_n, y_1, \ldots, y_l\}$ as depicted in Figures 3 and 4 respectively. Note that q_1 does not depend on the structure of φ and q_2 encodes every clause occurring in φ as a disjoint gadget. We will prove that Φ is satisfiable *iff* the CRPQ $q_1 \wedge q_2$ is bounded.

The construction is similar to the proof of (Figueira et al. 2020a, Theorem 4.3). However, the context is different since the cited theorem proves Π_2^p-hardness for the containment problem for the fragment of CRPQs where disjunctions of letters (*i.e.*, expressions like "$a + b$") are allowed.

Claim 19. *The Boolean query $q = y \xleftarrow{a} x \xrightarrow{a^*} z \xrightarrow{b} w$ is bounded. More precisely, $q \equiv \text{Exp}_1(q)$, i.e. q is equivalent to the disjunction of patterns $\bullet \xleftarrow{b} \bullet \xrightarrow{a} \bullet$ and $\bullet \xleftarrow{b} \bullet \xleftarrow{a} \bullet$.*

Note that this query q in Claim 19 is attached as a 'tail' for each of the x_i variable in the D gadget of q_1. As a consequence we have the following.

Claim 20. *q_1 is bounded.*

Theorem 18 is a consequence of the following lemma.

Lemma 21. *Let Φ be an instance of ∀∃-QBF and q_1 and q_2 as described. Then the following are equivalent: (1) Φ is satisfiable, (2) $q_1 \subseteq q_2$, and (3) $q_1 \wedge q_2$ is bounded.*

[6]Remember that $\ell > Z_q^+$ and we can add sufficiently many new vertices in the neighborhood of the vertex that is not in the image of h to obtain $x \xrightarrow{w^\ell} y$.

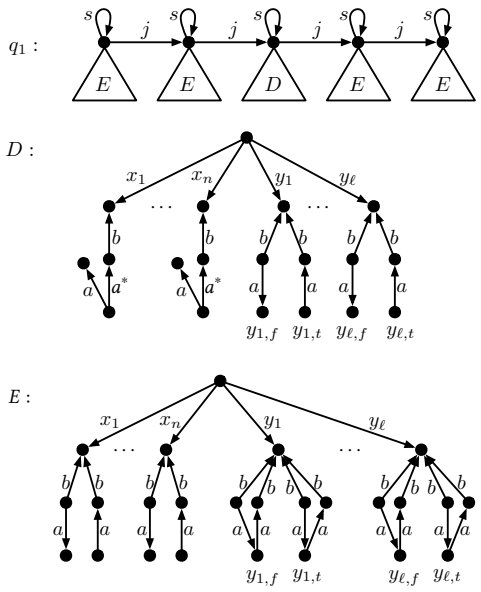

Figure 3: Query q_1 used in the proof of Theorem 18. Variables with identical $y_{i,\alpha}$ label of the gadgets D and E (across all E) represent the *same* variable (*e.g.*, $y_{1,f}$ in D and E are the same variable).

Proof. (1) ⇔ (2) (Proof sketch.)[7] This follows by an adaptation of the proof of Theorem 4.3 of (Figueira et al. 2020a). [8] The proof goes by reduction from ∀∃-QBF satisfiability. Given a ∀∃-QBF formula Φ as in (1), we construct queries q_1 and q_2 as depicted in Figures 3 and 4. The query q_2 consists of gadgets as in Figure 4 per clause, while q_1 is defined in Figure 3.

The "tail parts" of the edges labeled x_i in the D gadget allows any assignment to the variable x_i, that is, the a^* in D can be expanded for a 'true' assignment $\bullet \xleftarrow{b} \bullet \xleftarrow{a} \bullet$ or shrunk for a 'false' assignment $\bullet \xleftarrow{b} \bullet \xrightarrow{a} \bullet$. The valuation for the y_i variables comes from q_2 (as in Figure 4): the $y_{i,tf}$ node in q_2 embeds uniquely into one of the $y_{i,t}$ or $y_{i,f}$ nodes of q_1 to witness the containment.

When the formula is satisfiable, there exists a valuation for all the y_i variables for any valuation of the x_i variables. The D gadgets allows the choice of any true/false assignments for x_i variables, and we can embed the $y_{-,tf}$ nodes of q_2 into exactly one of the $y_{-,t}$, $y_{-,f}$ nodes of q_1 (depending on whether the y_i takes on true or false to make the formula true). Thus, this gives the containment of q_1 in q_2.

For the converse, assume the containment. Then we have an embedding of each clause of q_2 into the given graph database for a given choice of "tail parts" of the x_i labeled edges of D. In particular, we can always map a literal in each

[7]For a formal proof along similar lines, we refer the reader to (Figueira et al. 2020b, Theorem 3)

[8]Indeed, it suffices to reproduce the cited proof by replacing each occurrence of $\bullet \xrightarrow{t} \bullet$ with $\bullet \xleftarrow{b} \bullet \xleftarrow{a} \bullet$, and each occurrence of $\bullet \xrightarrow{f} \bullet$ with $\bullet \xleftarrow{b} \bullet \xrightarrow{a} \bullet$.

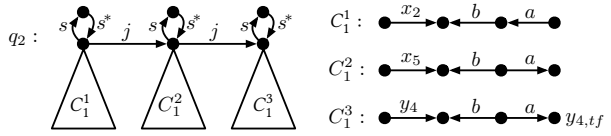

Figure 4: An example for query q_2, used in the proof of Theorem 18, for the clause $(x_2 \vee \neg x_5 \vee \neg y_4)$. There will be one such gadget for every clause of the formula in q_2. Variables having identical $y_{i,tf}$-label represent the *same* variable. Only the final variable of paths representing y_i-variables from Φ may have a $y_{i,tf}$-label.

clause of q_2 to D: this ensures satisfiability of the formula.

(2) $\Leftrightarrow$ (3) For the left-to-right direction, suppose $q_1 \subseteq q_2$. Note that we always have $q_1 \wedge q_2 \subseteq q_1$ and since $q_1 \subseteq q_2$ we also have $q_1 \subseteq q_1 \wedge q_2$. Thus, $q_1 \wedge q_2 \equiv q_1$. From Claim 20, q_1 is bounded and hence $q_1 \wedge q_2$ is also bounded.

For the right-to-left direction, assume $Q = q_1 \wedge q_2$ is bounded. In view of Proposition 2, it is equivalent to $\mathrm{Exp}_m(Q)$, for some m. We show that this implies $q_1 \subseteq q_2$. By contradiction, assume there is an expansion λ_1 of q_1 such that for all expansions λ_2 of q_2, λ_2 does not homomorphically map to λ_1.

Now consider the expansion λ_2 of q_2 in which all s^* atoms are expanded exactly $m+1$ times —*i.e.*, $\bullet \xrightarrow{s^*} \bullet$ is replaced with an s^{m+1}-path and hence we get a loop on s^{m+2} at each 'base' node in q_2.

Let $\lambda = \lambda_1 \wedge \lambda_2$, which is an expansion of Q. Since Q is equivalent to $\mathrm{Exp}_m(Q)$, we have $\lambda \subseteq \mathrm{Exp}_m(Q)$. In other words, there is an expansion $\hat{\lambda}$ of $\mathrm{Exp}_m(Q)$ and a homomorphism $h : \hat{\lambda} \xrightarrow{hom} \lambda$. Note that $\hat{\lambda}$ must be of the form $\hat{\lambda} = \hat{\lambda}_1 \wedge \hat{\lambda}_2$, where $\hat{\lambda}_1$ is an expansion of q_1 and $\hat{\lambda}_2$ is an expansion of q_2 where every s-cycle in $\hat{\lambda}_2$ is of length at most $m+1$.

But then $\hat{\lambda}_2 \xrightarrow{hom} \lambda_1$ is not possible by the choice of λ_1. This implies that we have $\hat{\lambda}_2 \xrightarrow{hom} \lambda_2$ which means that an s-cycle of length $\leq m+1$ is homomorphically mapped to an s-cycle of length $m+2$, which is not possible. Hence, no such homomorphism h can exist, which is a contradiction. □

7 Boundedness by Letter

If a query q in UCRPQ(SSF, $\mathbf{a}^*$) is not bounded, we can still try to bound it in as many atoms as possible, this is the object of the A-BOUNDEDNESS problem. Recall that $q \in$ UCRPQ($\mathbf{a}, \mathbf{a}^*$) is A-bounded if it is equivalent to a UCRPQ($\mathbf{a}, \mathbf{a}^*_{\bar{A}}$) query, with $\bar{A} = \mathbb{A} \setminus A$.

For $A \subseteq \mathbb{A}$ and $m \in \mathbb{N}$, let $q(A, m)$ be the result of replacing each a^* s.t. $a \in A$ with $a^{\leq m}$. We still have the characterization along the lines of Proposition 2.

Proposition 22. *$q \in$ CRPQ(SSF, $\mathbf{a}^*$) is A-bounded if, and only if, q is equivalent to $q(A, m)$ for some $m \in \mathbb{N}$.*

In fact, there is a unique maximal $A \subseteq \mathbb{A}$ such that q is A-bounded because of the following property.

Theorem 23. *If a query $q \in$ UCRPQ($\mathbf{a}, \mathbf{a}^*$) is A-bounded and B-bounded then q is also $(A \cup B)$-bounded.*

Proof idea. For any $A \subseteq \mathbb{A}$, $n \in \mathbb{N}$ and UCRPQ($\mathbf{a}, \mathbf{a}^*$) query q, let $q[A \mapsto n]$ be the result of replacing in q every $\xrightarrow{a^*}$ with $\xrightarrow{a^{\leq n}}$ for every $a \in A$. Hence, by Proposition 22, q is A-bounded if it is equivalent to $q[A \mapsto n]$ for some n. Let $q[A \mapsto n, B \mapsto m]$ denote $(q[A \mapsto n])[B \mapsto m]$.

From Proposition 22 let q be equivalent to both $q(A, n)$ and $q(B, m)$ and let $N_q = \max(n, m)$. Since q is A-bounded and B-bounded, we have $q[A \mapsto N_q] \equiv q[B \mapsto N_q] \equiv q$. It suffices to show that q is contained in $q[A \mapsto N_q, B \mapsto N_q \cdot |q|_{\mathsf{at}}]$. We will show that for every expansion λ of q there is an expansion λ' of $q[A \mapsto N_q, B \mapsto N_q \cdot |q|_{\mathsf{at}}]$ that maps homomorphically to λ.

Take then such λ. Since λ is contained in $q[B \mapsto N_q]$, there is an expansion λ_B of $q[B \mapsto N_q]$ which maps to λ. Since $q[B \mapsto N_q]$ is contained in $q[A \mapsto N_q]$, there is an expansion λ_A of $q[A \mapsto N_q]$ which maps homomorphically to λ_B. Pick such a λ_A of minimal size. Hence, we have the homomorphisms $h_1 : \lambda_A \xrightarrow{hom} \lambda_B$ and $h_2 : \lambda_B \xrightarrow{hom} \lambda$.

We next show that λ_A is in fact an expansion of $q[A \mapsto N_q, B \mapsto N_q \cdot |q|_{\mathsf{at}}]$, which would already prove the statement. By means of contradiction, if it is not the case, there is some B-atom expansion of length $m > N_q \cdot |q|_{\mathsf{at}}$ in λ_A.

This means that the h_1-image of such a B-atom expansion induces a cycle in λ_B. We can then "cut out" the cycle producing another expansion λ'_A of $q[A \mapsto N_q]$ which is strictly smaller than λ_A and which still maps homomorphically to λ_B via h_1 (restricted to λ'_A). This contradicts the minimality of λ_A. □

Towards proving the upper bound of Theorem 7, Lemma 14 and Lemma 15 can be generalized trivially:

Lemma 24. *Let $q \in$ UCRPQ($\mathbf{a}, \mathbf{a}^*$) and $A \subseteq \mathbb{A}$. If q is A-bounded, then it is equivalent to $q(A, Z_q)$ for $Z_q \hat{=} |q|^3_{\mathsf{at}} \cdot N_q \cdot |q|_{\mathsf{var}} \cdot \Pi_{w \in \mathcal{R}_q} |w|$.*

Lemma 25. *Let $q \in$ UCRPQ($\mathbf{a}, \mathbf{a}^*$) and $A \subseteq \mathbb{A}$. Then q is A-bounded iff $q(A, Z_q)$ is equivalent to $q(A, Z_q^+)$, for $Z_q^+ \hat{=} |q|_{\mathsf{at}} \cdot Z_q + 1$.*

The proofs of both statements follow along the same lines as those of Lemma 14 and Lemma 15 respectively, except that we need to work with $q(A, m)$ expansions instead of $q(m)$. Note that in the proof, the operations of extending and contracting of paths in the recursive atom expansions are done only for only those a^* atoms such that $a \in A$. For all $b \notin A$ we keep the atom expansion paths unaltered and hence the arguments continue to hold. Together, the two statements above give us the following upper bound.

Theorem 26. *The A-BOUNDEDNESS problem for UCRPQ($\mathbf{a}, \mathbf{a}^*$) is in Π^p_2. Further, an equivalent UCRPQ($\mathbf{a}, \mathbf{a}^*_{\bar{A}}$) query of linear size can be computed, with $\bar{A} = \mathbb{A} \setminus A$.*

Proof of Theorem 7. Item (1) The upper bound follows from Theorem 26 and lower bound from Theorem 18.

Item (2) The existence of a unique maximal A follows from Theorem 23. In order to find it, a Π_2^p algorithm can check if q is $\{a\}$-bounded for every $a \in \mathbb{A}$, and we know, thanks to Theorem 23, that the maximal A will be the set of all the a's for which q turned out to be $\{a\}$-bounded. The equivalent A-bounded query is then $q(A, Z_q)$. □

8 Conclusion

We have shown that the BOUNDEDNESS problem for UCRPQs with simple recursion of the form a^* is Π_2^p-complete. This is in line with the complexity for the containment problem of similar fragments (Figueira et al. 2020a).

Constants and free variables. While we have focused our study on Boolean queries without constants, our upper bounds also extend to queries containing constants and free variables via the classical notion of homomorphism between conjunctive queries with constants and free variables. For the case of free variables, one can use a standard reduction to Boolean queries: For any formula with free variables, consider replacing each free variable x with a bound variable y_x and adding a self-loop $y_x \xrightarrow{a_x} y_x$, where a_x is a fresh alphabet symbol depending on x. It follows that the original query is bounded *iff* the modified Boolean query is bounded.

Less trivial recursion. Instead of allowing just for one word or one letter to appear under a Kleene star, as in a^*, one could also consider the case of a set of letters appearing under a Kleene star, as in $(a+b+c)^*$. The simplest class of regular expressions containing such behavior, namely the one containing only the regular expression a for each $a \in \mathbb{A}$ and A^* for any set $A \subseteq \mathbb{A}$, is denoted by $(\mathbf{a}, \mathbf{A}^*)$ in (Figueira et al. 2020a). The containment problem for CRPQ($\mathbf{a}$, $\mathbf{A}^*$) does not enjoy a better complexity than the containment problem for general CRPQ, that is, it is ExpSpace-complete.

We would expect a similar behaviour for the BOUNDEDNESS problem of UCRPQ(SSF, $\mathbf{A}^*$) but the complexity checking if such a query is bounded is open. The ExpSpace upper bound follows from Theorem 3, but no matching lower bound is known. We can also generalize this question to UCRPQ(SSF, $\mathbf{W}^*$) where $\mathbf{W}^*$ denotes the regular expressions of the form $(w_1 + \ldots + w_k)^*$.

Open problem 27. *What is the complexity for the* BOUNDEDNESS *problem for* UCRPQ(SSF, $\mathbf{A}^*$) *and* UCRPQ(SSF, $\mathbf{W}^*$)*?*

Rewritability of ontology-mediated CRPQs. CRPQs can also be considered in the presence of an ontology (see, *e.g.*, (Baget et al. 2017; Gutiérrez-Basulto, Ibáñez-García, and Jung 2018)). In such a context, an ontology-mediated query (OMQ) (T, q) admits a rewriting in a language $\mathcal{L}$ if there is a query $q' \in \mathcal{L}$ such that for every ABox A we have that A, T entails q if, and only if, $A \models q'$. For Horn description logics such as $\mathcal{ELHI}_\bot$, the resulting OMQs (T, q) (where $T \in \mathcal{ELHI}_\bot$ and $q \in$ CRPQ) is closed under homomorphisms. In these cases the OMQ is FO-rewritable iff it is UCQ-rewritable. Hence, our positive complexity result can be seen as a first step towards investigating the rewritability of OMQs with CRPQs as query langauges.

Acknowledgements

Diego Figueira is partially supported by ANR AI Chair INTENDED, grant ANR-19-CHIA-0014.

References

Angles, R.; Arenas, M.; Barceló, P.; Hogan, A.; Reutter, J. L.; and Vrgoc, D. 2017. Foundations of modern query languages for graph databases. *ACM Comput. Surv.* 50(5):68:1–68:40.

Baeza, P. B. 2013. Querying graph databases. In *ACM Symposium on Principles of Database Systems (PODS)*, 175–188. ACM Press.

Baget, J.; Bienvenu, M.; Mugnier, M.; and Thomazo, M. 2017. Answering conjunctive regular path queries over guarded existential rules. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 793–799. ijcai.org.

Barceló, P.; Figueira, D.; and Romero, M. 2019. Boundedness of conjunctive regular path queries. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 104:1–104:15. Leibniz-Zentrum für Informatik.

Bienvenu, M.; Hansen, P.; Lutz, C.; and Wolter, F. 2016. First order-rewritability and containment of conjunctive queries in horn description logics. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 965–971. IJCAI/AAAI Press.

Bonifati, A.; Martens, W.; and Timm, T. 2019. Navigating the maze of wikidata query logs. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 127–138. ACM.

Calvanese, D.; Giacomo, G. D.; Lenzerini, M.; and Vardi, M. Y. 2000. Containment of conjunctive regular path queries with inverse. In *Principles of Knowledge Representation and Reasoning (KR)*, 176–185.

Figueira, D., and Morvan, R. 2023. Approximation and semantic tree-width of conjunctive regular path queries. In *International Conference on Database Theory (ICDT)*, volume 255 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 15:1–15:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Figueira, D.; Godbole, A.; Krishna, S.; Martens, W.; Niewerth, M.; and Trautner, T. 2020a. Containment of simple conjunctive regular path queries. In *Principles of Knowledge Representation and Reasoning (KR)*, 371–380.

Figueira, D.; Godbole, A.; Krishna, S.; Martens, W.; Niewerth, M.; and Trautner, T. 2020b. Containment of simple regular path queries. *CoRR* abs/2003.04411.

Florescu, D.; Levy, A. Y.; and Suciu, D. 1998. Query containment for conjunctive queries with regular expressions. In *ACM Symposium on Principles of Database Systems (PODS)*, 139–148. ACM Press.

Gutiérrez-Basulto, V.; Ibáñez-García, Y. A.; and Jung, J. C. 2018. Answering regular path queries over SQ ontologies. In *Proceedings of AAAI Conference on Artificial Intelligence*, 1845–1852. AAAI Press.

Haase, C.; Kreutzer, S.; Ouaknine, J.; and Worrell, J. 2009. Reachability in succinct and parametric one-counter automata. In *International Conference on Concurrency Theory (CONCUR)*, volume 5710 of *Lecture Notes in Computer Science*, 369–383. Springer.

Kopczyński, E., and To, A. W. 2010. Parikh images of grammars: Complexity and applications. In *Annual Symposium on Logic in Computer Science (LICS)*, 80–89. IEEE Computer Society Press.

Malyshev, S.; Krötzsch, M.; González, L.; Gonsior, J.; and Bielefeldt, A. 2018. Getting the most out of wikidata: Semantic technology usage in wikipedia's knowledge graph. In *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II*, volume 11137 of *Lecture Notes in Computer Science*, 376–394. Springer.

Martens, W.; Neven, F.; and Schwentick, T. 2009. Complexity of decision problems for XML schemas and chain regular expressions. *SIAM Journal on computing* 39(4):1486–1530.